Supplemental Material

Supplement for Neural networks for post-processing ensemble weather forecasts

Stephan Rasp[1] and Sebastian Lerch[2,3]

[1]Meteorological Institute, Ludwig-Maximilians-Universität, Munich
[2]Institute for Stochastics, Karlsruhe Institute of Technology
[3]Heidelberg Institute for Theoretical Studies

August 9, 2018

1 Calibration assessment

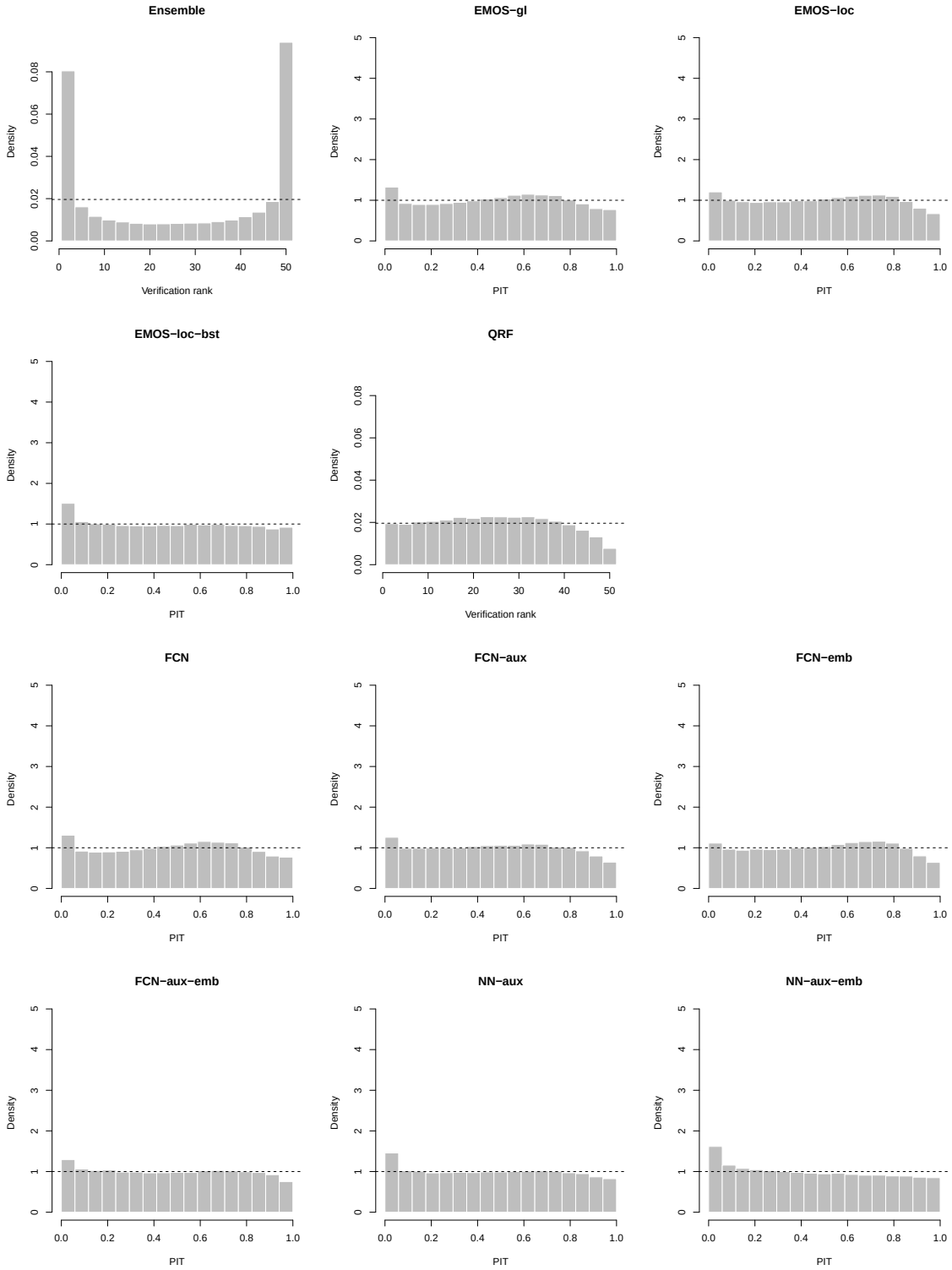

Figure 1: Verification rank and PIT histograms for raw and post-processed ensemble forecasts based on models estimated using data from 2015, aggregated over all forecast cases during the evaluation period in calendar year 2016.

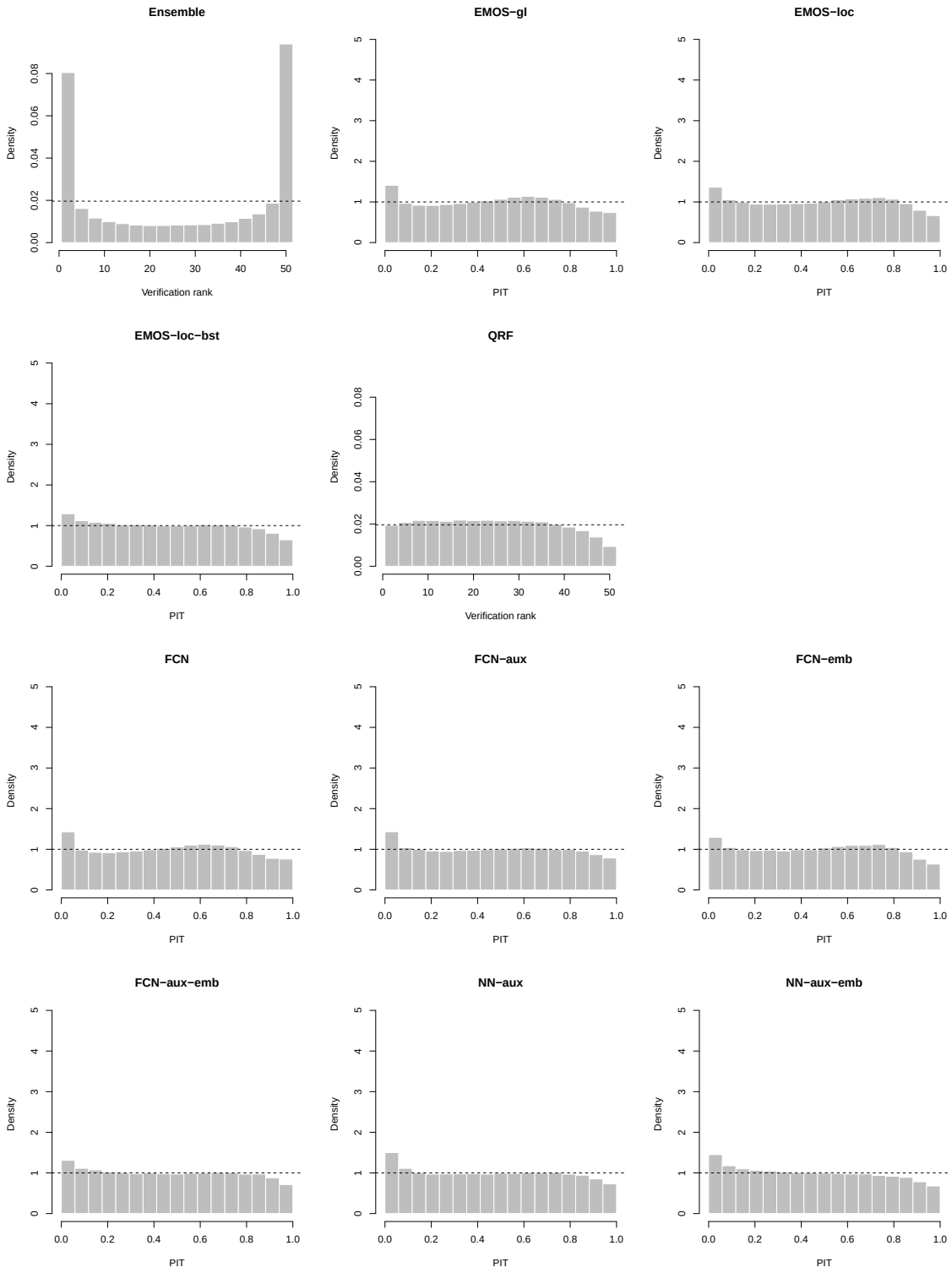

Figure 2: Verification rank and PIT histograms for raw and post-processed ensemble forecasts based on models estimated using data from 2007–2015, aggregated over all forecast cases during the evaluation period in calendar year 2016..

2 CRPSS results for alternative benchmark models

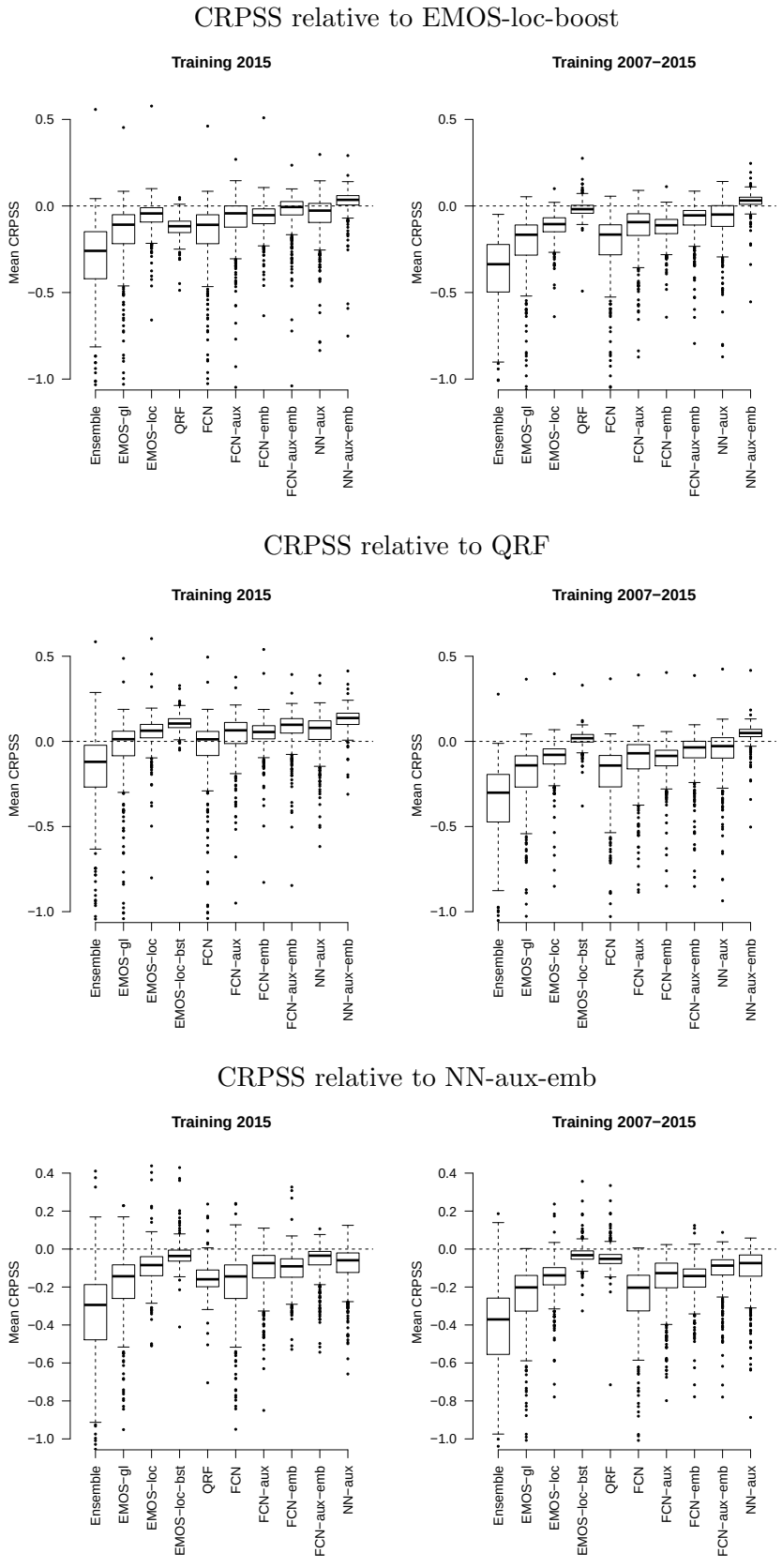

Figure 3: As Figure 3, but with different reference models.

3 Details on computational aspects

Table 1 shows computation times required for training the different post-processing models for both training sets. As noted before, the computation times are not directly comparable due to implementations in different programming languages and hardware environments. The computation times for the benchmark models, implemented in `R` using the `crch` (Messner et al., 2016), `quantregForest` (Meinshausen, 2017) and `scoringRules` (Jordan et al., 2018) packages, were obtained on a standard laptop computer, whereas the network models were implemented with the `Python` libraries Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2016), and run on a single GPU (Nvidia Tesla K20). Computation times on a regular CPU are roughly 6 times longer for the most complex networks. For the simple networks the difference is negligible. Note that the inference time, i.e., the time to make a prediction after the model has been trained, is on the order of a few seconds for all models. Further, note that all computation times reported here are substantially lower compared to the computational costs of generating the raw ensemble forecast.

Tables 2 and 3 list hyperparameters for the benchmark and network models.

Table 1: Computation times (in minutes) for estimating post-processing models with the two training sets and computing out-of-sample forecasts for the evaluation period.

Model	Computation time (min) with training data from	
	2015	2007–2015
Benchmark models		
EMOS-gl	< 1	< 1
EMOS-loc	< 1	1
EMOS-loc-bst	14	48
QRF	8	430
Network models		
FCN	< 1	1
FCN-aux	< 1	2
FCN-emb	< 1	3
FCN-aux-emb	< 1	3
NN-aux	4	25
NN-aux-emb	9	16

Table 2: Hyperparameters for benchmark models. AIC denotes the Akaike information criterion.

Model	Parameter	Value
EMOS-gl	none	
EMOS-loc	none	
EMOS-loc-bst	maximum number of iterations	1 000
	step size	0.05
	stopping criterion for boosting algorithm	AIC
QRF	number of trees	1 000
	minimum size of terminal leaves	10
	number of variables randomly sampled as candidates at each split	25

Table 3: Hyperparameters for network models. Values in parentheses indicate settings for the longer training period from 2007–2015. Parameters refers to all learnable values: weights, biases and latent embedding features. An epoch refers to one pass through all training samples. Batch size refers to the number of random training samples considered per gradient update in the SGD optimization.

Model	Number of parameters	Epochs	Learning rate	Batch size	Hidden nodes	Embedding size
FCN	6	30 (15)	0.1 (0.1)	4 096 (4 096)		
FCN-aux	82	30 (10)	0.02 (0.02)	1024 (1 024)		
FCN-emb	1 084	30 (10)	0.02 (0.02)	1 024 (1 024)		2 (2)
FCN-aux-emb	1 160	30 (10)	0.02 (0.02)	1 024 (1 024)		2 (2)
NN-aux	3 326	(10)	(0.02)	(1 024)	(64)	(2)
NN-aux-emb	24 116	30 (10)	0.01 (0.002)	1 024 (4 096)	50 (512)	2 (2)

4 Statistical significance of score differences

Pair-wise one-sided Diebold-Mariano tests are applied to all possible comparisons of forecast models at each of the 499 stations individually. To account for multiple hypothesis testing and spatial correlations of score differences, we apply a Benjamini-Hochberg procedure to the corresponding p-values when aggregating the results by determining the ratio of stations with significant score differences, see Appendix **??** for details.

Table 4 summarizes pair-wise Diebold-Mariano tests by showing the ratio of stations with statistically significant CRPS differences after applying a Benjamini-Hochberg procedure for a nominal level of $\alpha = 0.05$. Generally, the results indicate large numbers of stations with significant differences of the network models when compared to standard EMOS approaches. NN-aux-emb shows the highest ratios of significant score differences over any competitor, and is significantly outperformed at very few station and only by the best-performing alternatives.

Table 4: Ratio of stations (in %) where pair-wise Diebold-Mariano tests indicate statistically significant CRPS differences after applying a Benjamini-Hochberg procedure to account for multiple testing for a nominal level of $\alpha = 0.05$ of the corresponding one-sided tests. The (i, j)-entry in the i-th row and j-th column indicates the ratio of stations where the null hypothesis of equal predictive performance of the corresponding one-sided Diebold-Mariano test is rejected in favor of the model in the i-th row when compared to the model in the j-th column. The remainder of the sum of (i, j)- and (j, i)-entry to 100% is the ratio of stations where the score differences are not significant.

Training with 2015 data

	Ens.	EMOS -gl	EMOS -loc	EMOS -loc-bst	QRF	FCN	FCN -aux	FCN -emb	FCN -aux-emb	NN -aux	NN -aux-emb
Ens.		0.6	0.0	0.0	0.6	0.6	0.8	0.0	0.8	0.8	0.6
EMOS-gl	83.2		0.2	0.0	10.4	10.2	3.0	0.2	0.6	2.0	0.2
EMOS-loc	96.2	71.3		0.0	50.5	71.9	17.4	24.8	5.2	9.6	1.4
EMOS-loc-bst	93.8	72.7	40.5		89.8	74.3	41.7	49.1	21.0	30.5	2.0
QRF	54.7	22.0	3.6	0.0		22.4	8.0	3.6	3.4	5.2	0.2
FCN	83.0	7.4	0.2	0.0	10.4		3.0	0.2	0.6	2.0	0.2
FCN-aux	83.2	60.3	17.2	1.8	47.5	62.3		19.0	1.0	0.4	0.2
FCN-emb	89.4	67.1	1.0	0.0	44.1	68.1	11.4		0.8	6.4	0.6
FCN-aux-emb	86.6	78.8	53.1	7.6	69.1	79.6	55.1	58.5		27.1	0.2
NN-aux	87.2	69.5	25.9	2.0	57.5	70.7	22.8	30.9	8.0		0.4
NN-aux-emb	93.6	89.4	67.1	30.3	92.2	90.2	67.3	72.7	43.5	64.9	

Training with 2007-2015 data

	Ens.	EMOS -gl	EMOS -loc	EMOS -loc-bst	QRF	FCN	FCN -aux	FCN -emb	FCN -aux-emb	NN -aux	NN -aux-emb
Ens.		0.6	0.0	0.0	0.0	0.6	0.8	0.0	0.8	0.8	0.0
EMOS-gl	86.8		0.2	0.0	0.2	2.6	3.0	0.2	0.6	0.2	0.0
EMOS-loc	98.8	72.7		0.0	0.2	71.7	17.2	17.4	3.6	6.8	0.6
EMOS-loc-bst	99.4	98.0	91.4		21.0	97.8	82.0	94.2	70.3	49.7	1.4
QRF	98.6	94.2	79.2	1.4		94.2	57.7	84.4	38.1	33.5	1.2
FCN	87.8	11.0	0.2	0.0	0.2		3.2	0.2	0.6	0.2	0.0
FCN-aux	87.6	65.5	24.2	0.0	0.4	65.5		26.7	0.8	1.4	0.0
FCN-emb	93.4	71.3	0.0	0.0	0.2	70.5	12.0		1.2	4.6	0.0
FCN-aux-emb	91.2	82.8	60.3	0.0	0.6	81.8	58.1	64.1		16.4	0.0
NN-aux	95.6	84.8	54.5	1.4	9.8	84.8	72.9	58.5	34.5		0.0
NN-aux-emb	98.8	97.8	95.2	29.9	52.9	97.6	92.0	96.0	91.0	74.5	

References

Abadi, M., and Coauthors, 2016: Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Chollet, F., and Coauthors, 2015: Keras. https://keras.io.

Jordan, A., F. Krüger, and S. Lerch, 2018: Evaluating probabilistic forecasts with scoringRules. *Journal of Statistical Software*, in press, preprint available at https://arxiv.org/abs/1709.04743.

Meinshausen, N., 2017: *quantregForest: Quantile Regression Forests*. URL https://CRAN.R-project.org/package=quantregForest, r package version 1.3-7.

Messner, J. W., G. J. Mayr, and A. Zeileis, 2016: Heteroscedastic censored and truncated regression with crch. *The R Journal*, **8**, 173–181.